

12/. Viết chương trình cho người dùng nhập 1 số nguyên dương n. sắp xếp các số trong n thành từ nhỏ đến lớn (nMin) và từ lớn đến nhỏ (nMax). Tìm hiệu của nMax và nMin.

Ví dụ nhập n=1423

$$\Rightarrow nMin=1234, \quad nMax=4321, \quad nMax - nMin = 4321 - 1234 = 3087$$

☞ **Gợi ý:** Đưa từng số trong n vào list. Sử dụng các phương thức join và sort của dữ liệu chuỗi để tìm nMin và nMax, từ đó suy ra kết quả hiệu

Project

Buoi07 D:\TTTH_KHTN\Python270\Buoi07

Bai12_Tr28.py

temp.py

External Libraries

Scratches and Consoles

temp.py

Bai12_Tr28.py

```
1 lst=list(input("Nhập 1 số nguyên"))
2 print(lst)
3 nho= int("".join(sorted(lst)))
4 lon= int("".join(sorted(lst, reverse=True)))
5 print(lon, " - ", nho, " = ", lon-nho)
```

Bai12_Tr28

Nhập 1 số nguyên>? 729461

['7', '2', '9', '4', '6', '1']

976421 - 124679 = 851742

In[3]:

15/. Tạo 1 list gồm n số nguyên với giá trị ngẫu nhiên từ 0 đến 9. Viết 2 hàm nhận tham số là list vừa tạo, 1 hàm thực hiện vẽ biểu đồ (histogram) ngang và 1 hàm thực hiện vẽ biểu đồ dọc. Biết biểu đồ được vẽ bằng dấu hoa thị (*). Minh họa:

$n=4$

list=[2, 0, 7, 3]

*

*

*

*

*

*

*

*

*

*

*

*

*

*

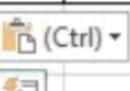
Biểu đồ ngang

Biểu đồ dọc

A B C D E F G H I J K L M N O P Q R S T U V W X

4 2 0 9 3 dòng

ở đây có 5 số
nên có 5 cột. Nếu
có n số thì sẽ có
 n cột



```
temp.py X Bai15_Tr29.py X Bai12_Tr28.py X
2 def NhapSo():
3     while(True):
4         n=int(input("Nhập số lượng phần tử của list: "))
5         if n<=0:
6             print("Chỉ nhận các số >0. Nhập lại")
7         else:
8             return n
9 #-----
10 def TaoList(n):
11     lst=[]
12     for i in range(n):
13         lst.append(random.randint(0,9))
14     return lst
15
TaoList()
```

temp.py

Bai15_Tr29.py

Bai12_Tr28.py

```
5         if  n<=0:
6             print("CHỉ nhận các số >0. Nhập lại")
7         else:
8             return n
9
#-----
10 def TaoList(n):
11     lst=[]
12     for i in range(n):
13         lst.append(random.randint(0,9))
14     return lst
15
#-----
16 def BieuDoDoc(lst):
17     |
18 ====== CHUONG TRINH CHINH ======
19 n= NhapSo()
20 lst=TaoList(n)
21 BieuDoDoc(lst)
22 #BieuDoDoc(TaoList(NhapSo()))
```

A B C D E F G H I J K L M N O P Q R S T U V W

0 1 2 3 4

4 | 2 | 0 | 9 | 3

dòng

0 1 2 3 4 5 6 7 8 9 10

0

1

2

3

4

5

6

7

8

9

X

X

X

X

X

X

X

X

X

BuoI07 > Bai15_Tr29.py

Project + - temp.py Bai15_Tr29.py Bai12_Tr28.py

BuoI07 D:\TTTH_KHTN\Python270\BuoI07
 Bai12_Tr28.py
 Bai15_Tr29.py
 temp.py

> External Libraries
Scratches and Consoles

```
14     return lst
15
16 def BieuDoDoc(lst):
17     for dong in range(10): # dong di tu 0-9
18         for cot in range(len(lst)):
19             if lst[cot]>dong:
20                 print("* ", end="")
21             print()
22     ====== CHUONG TRINH CHINH ======
23     n= NhapSo()
24     lst=TaoList(n)
25     print("List vừa tao: ",lst)
```

BieuDoDoc() > for dong in range (10) > for cot in range(len(lst))

Bai12_Tr28 Bai15_Tr29

Nhập số lượng phần tử của list: >? 6

List vừa tạo: [0, 9, 0, 5, 5, 5]

* * * * |
* * * *
* * * *
* * * *

In[3]:

Ist = [
0, 9, 0, 5, 5, 5]
n = {i:
Special

A6

f_x

dong+giatri>=10

temp.py

Bai15_Tr29.py

Bai12_Tr28.py

return lst

#-----

def BieuDoDoc(lst):

for dong in range(10): # dong di tu 0-9

for cot in range(len(lst)):

if dong+lst[cot]>=10:

print("* ", end="")

else:

print(" ", end="")

print()

#=====

CHUONG TRINH CHINH

=====

n= NhapSo()

BieuDoDoc() > for dong in range(10) > for cot in range(len(lst)) > else

Buoi07 > Bai15_Tr29.py

1: Project

Project



temp.py

Bai15_Tr29.py

Bai12_Tr28.py

Buoi07 D:\TTTH_KHTN\Python270\Buoi07
Bai12_Tr28.py
Bai15_Tr29.py
temp.py
External Libraries
Scratches and Consoles

```
20 print("* ", end="")
21
22 else:
23     print(" ", end="")
24 print("\n")
BieuDoDoc() > for dong in range(10) > for cot in range(len(lst)) > else
```

Bai15_Tr29

Nhập số lượng phần tử của list: >? 6
List vừa tạo: [3, 9, 3, 8, 3, 5]



```
*
```

```
* *
```

```
* *
```

```
*
```

```
* * *
```

```
* * *
```

```
* * * * *
```

```
* * * * *
```

```
* * * * *
```

đã đúng

In[3]:

2: Favorites

ucture

Buoi07 > Bai15_Tr29.py

Project

Buoi07 D:\TTTH_KHTN\Python270\Buoi07

Bai12_Tr28.py

Bai15_Tr29.py

temp.py

External Libraries

Scratches and Consoles

```
temp.py × Bai15_Tr29.py × Bai12_Tr28.py ×
9  -----
10 def TaoList(n):
11     lst=[]
12     for i in range(n):
13         lst.append(random.randint(0,9))
14     return lst
15 -----
16 def BieuDoDoc(lst):
17     for dong in range (10): # dong di tu 0-9
18         for cot in range(len(lst)):
19             if dong+lst[cot]>=10:
20                 print("* ", end="")
21             else:
22                 print(" ", end="")
23         print()
24 ===== CHUONG TRINH CHINH =====
25 n= NhapSo()
26 lst=TaoList(n)
27 print("List vừa tạo: ",lst)

BieuDoDoc() > for dong in range (10) > for cot in range(len(lst)) > else
```

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

0 1 2 3 4

4 2 0 9 3

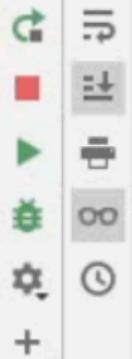
dòng

0 1 2 3 4 5 6 7 8 9 10

0	x	x	x	x																		
1	x	x																				
2																						
3	x	x	x	x	x	x	x	x	x	x												
+	x	x	x																			

biểu đồ ngang

```
temp.py × Bai15_Tr29.py × Bai12_Tr28.py ×
24                                     print()
25
26 def BieuDoNgang(lst):
27     print("In biểu đồ ngang")
28     for dong in range(len(lst)):
29         for cot in range(lst[dong]):
30             print("* ", end="")
31         print()
32 #===== CHUONG TRINH CHINH =====
33 n= NhapSo()
34 lst=TaoList(n)
35 print("List vừa tạo: ",lst)
36 BieuDoDoc(lst)
37 #BieuDoDoc(TaoList(NhapSo()))
38 BieuDoNgang(lst)
39
40
```



In biểu đồ ngang

```
* *
* * * * * *
* * * * *
* *
* * * * * * *
```

In[3]:

23/. Viết chương trình cho người dùng nhập 1 chuỗi (S) và ký tự dùng để phân chia chuỗi (c). Thực hiện phân chia chuỗi S thành các phần sao cho các phần đó không còn chứa các ký tự

Python Exercises

dùng để phân chia c. Cho biết số lượng chuỗi sau khi phân tách (không kể những chuỗi rỗng ("").

87

Yêu cầu thực hiện bằng 2 cách:

- Cách 1: thực hiện đếm khi không xóa chuỗi rỗng ra khỏi list.
- Cách 2: thực hiện đếm sau khi đã xóa chuỗi rỗng ra khỏi list.

Ví dụ: với `s='https://www.w3resource.com/python-exercises/string' và c='/'`

BuoI07 > Bai23_Tr30.py

Project ▾ 1: Project

BuoI07 D:\TTTH_KHTN\Python
Bai12_Tr28.py
Bai15_Tr29.py
Bai23_Tr30.py
temp.py
External Libraries
Scratches and Consoles

temp.py × Bai23_Tr30.py ×

```
def Cach1(S,ch):
    lst=S.split(ch)
    print(lst)
    dem=lst.count('')
    print("Số lượng chuỗi rỗng có trong list =",dem)
    print("Số lượng chuỗi sau khi phân tách (không kể chuỗi rỗng)=", len(lst)-dem)

#-----CHUONG TRINH CHINH -----
S = 'https://www.w3source.com/python-exercises/string'
ch='/'
Cach1(S,ch)
```

Bai15_Tr29 × Bai23_Tr30 ×

Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]
['https:', '', 'www.w3source.com', 'python-exercises', 'string']
Số lượng chuỗi rỗng có trong list = 1
Số lượng chuỗi sau khi phân tách (không kể chuỗi rỗng)= 4

In[3]:

01 S = {str} 'https://www.w3source.c
01 ch = {str} '/'
Special Variables

Buoi07 > Bai23_Tr30.py

Bai

1: Project

- Buoi07 D:\TTTH_KHTN\Python:
 - Bai12_Tr28.py
 - Bai15_Tr29.py
 - Bai23_Tr30.py
 - temp.py
- External Libraries
- Scratches and Consoles

```
def Cach1(S,ch):
    lst=S.split(ch)
    print(lst)
    dem=lst.count('')
    print("Số lượng chuỗi rỗng có trong list =",dem)
    print("Số lượng chuỗi sau khi phân tách (không kể chuỗi rỗng)=", len(lst)-dem)
    #-----
def Cach2(S,ch):
    lst=S.split(ch)
    print(lst)
    # xóa tất cả các chuỗi rỗng có trong list
    rong=""
    while rong in lst:
        lst.remove(rong)
    print("List sau khi xóa tất cả các chuỗi rỗng:",lst)
    print("Số lượng chuỗi sau khi phân tách (không kể chuỗi rỗng)=", len(lst))
```

2: Favorites

Bai15_Tr29 < Bai23_Tr30 <

3: Structure

In[3]:



01 S = {str} 'https://www.w3source.co
01 ch = {str} '/'
> Special PyCharm 2021.2.3 a
Update

Buoi07 > Bai23_Tr30.py

Ba

1: Project

- Project
- Buoi07 D:\TTTH_KHTN\Python270\Buoi07
 - Bai12_Tr28.py
 - Bai15_Tr29.py
 - Bai23_Tr30.py
 - temp.py
- External Libraries
- Scratches and Consoles

```
temp.py x Bai23_Tr30.py x
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23

print(lst)
# xóa tất cả các chuỗi rỗng có trong list
rong=""
while rong in lst:
    lst.remove(rong)
print("List sau khi xóa tất cả các chuỗi rỗng:",lst)
print("Số lượng chuỗi sau khi phân tách (không kể chuỗi rỗng)=", len(lst))

-----CHUONG TRINH CHINH -----
S = 'https://www.w3source.com/python-exercises/string'
ch='/'
#Cach1(S,ch)
Cach2(S,ch)
```

2: Favorites

Bai15_Tr29 x Bai23_Tr30 x

Số lượng chuỗi rỗng có trong list = 1
Số lượng chuỗi sau khi phân tách (không kể chuỗi rỗng)= 4

Structure

- 01 S = {str} 'https://www.w3source.co...
- 01 ch = {str} '/'
- Special Variables

File Edit View Navigate Code Refactor Run Tools VCS Window Help Buoi07 [D:\TTTH_KHTN\Python270\Buoi07] - ...\\Bai23_Tr30.py - PyCharm (Administrator)

Buoi07 > Bai23_Tr30.py

Bai23_Tr30

Project ▾

Buoi07 D:\TTTH_KHTN\Python270\Buoi07

- Bai12_Tr28.py
- Bai15_Tr29.py
- Bai23_Tr30.py
- temp.py

External Libraries

Scratches and Consoles

temp.py x Bai23_Tr30.py x

```
9     lst = s.split(',')
10    print(lst)
11    # xóa tất cả các chuỗi rỗng có trong list
12    rong = ""
13    while rong in lst:
14        lst.remove(rong)
```

Cach20 > while rong in lst

Bai15_Tr29 x Bai23_Tr30 x

"C:\Program Files\Python39\python.exe" "C:\Program Files\JetBrains\PyCharm Community Edition 2019.3.4\plugins\

import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['D:\\TTTH_KHTN\\Python270\\Buoi07', 'D:/TTTH_KHTN/Python270/Buoi07'])

In[2]: runfile('D:/TTTH_KHTN/Python270/Buoi07/Bai23_Tr30.py')
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] In[2]: runfile('D:/TTTH_KHTN/Python270/Buoi07/Bai23_Tr30.py')
['https:', '', 'www.w3source.com', 'python-exercises', 'string']
List sau khi xóa tất cả các chuỗi rỗng: ['https:', 'www.w3source.com', 'python-exercises', 'string']
Số lượng chuỗi sau khi phân tách (không kể chuỗi rỗng)= 4

In[3]:



3.1.7. Hàm ẩn danh (Anonymous function - lambda)

3.1.7.1. Giới thiệu

- Gọi là *Anonymous function* vì function không được khai báo theo cách tiêu chuẩn bằng cách dùng từ khóa *def* mà được dùng 1 cách ngắn gọn bằng từ khóa *lambda*. Thường các *function* dạng này được viết chỉ trên 1 dòng lệnh.

3.1.7.2. Cú pháp

`lambda [argument1 [, argument2, ...]]: expression`

Giải thích:



- *lambda function*:
 - *Lambda function* có *local namespace* riêng và không thể truy cập các biến khác và các biến trong phạm vi *global namespace*.
 - Kết quả trả về có thể gồm 1 hoặc nhiều giá trị như các hàm bình thường.
- *argument*: Tương tự như các hàm thông thường, số lượng đối số truyền vào của *lambda* là không giới hạn.

- Có thể sử dụng tối đa 1 lệnh *if* trong expresion (xem ví dụ 3.19).
- Có thể gọi hàm khác trong expression (xem ví dụ 3.20).

3.1.7.3. Các cách xây dựng lambda function

- Trường hợp 1: sử dụng thay cho cách viết hàm thông thường

- Ví dụ 3.18: lambda không có đối số

Sử dụng Anonymous function

```
Dung = lambda: True
Sai = lambda: False
#--- CHƯƠNG TRÌNH CHÍNH ---
n=int(input('Nhập điểm: '))
print('Đã hoàn tất môn học: ',end=' ')
if n>=5:
    print(Dung())
else:
    print(Sai())
```

Kết quả

'''Tùy thuộc vào giá trị của n mà kết quả sẽ in ra 1 trong 2 dạng sau:
Đã hoàn tất môn học: True
Đã hoàn tất môn học: False
'''

2.2. Anonymous function (hàm ẩn danh)

2.2.1. Xây dựng hàm ẩn danh với đối tượng number

13/. Viết chương trình Python sử dụng lambda để tính cho các trường hợp sau:

- a) Hàm nhận 1 đối số là số nguyên n và trả về trị tuyệt đối của n.
- b) Hàm nhận 1 đối số là số nguyên n và trả về giá trị của $n+15$.
- c) Hàm nhận 2 đối số là số nguyên (x, y), trả về tích của x và y.
- d) Hàm nhận 1 đối số là số nguyên n. Cho biết n có là bội số của 13 hoặc 19 hay không?
- e) Hàm nhận 1 đối số là số thực r là bán kính của hình tròn. Cho biết diện tích, chu vi hình tròn.
- f) Hàm nhận 2 đối số là số thực d, r là chiều dài và chiều rộng của hình chữ nhật. Cho biết chu vi và diện tích hình chữ nhật.

☞ Gợi ý thực hiện câu e và f: hàm lambda trả về nhiều giá trị.

Buoi07 > Bai13_Tr21.py

Project ▾

Buoi07 D:\TTTH_KHTN\Python270\Buoi07

- Bai12_Tr28.py
- Bai13_Tr21.py
- Bai15_Tr29.py
- Bai23_Tr30.py
- temp.py

External Libraries

Scratches and Consoles

temp.py

Bai13_Tr21.py

```
CauA = lambda n: abs(n)
n=int(input("Nhập số"))
print("Câu A: Trị tuyệt đối của %d là %d" %(n,CauA(n)))
```

Bai15_Tr29 × Bai23_Tr30 × Bai13_Tr21 ×

Import

Run

Stop

Output

Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]

Nhập số? -7

Câu A: Trị tuyệt đối của -7 là 7

Buo107 > Bai13_Tr21.py

1: Project

- Buo107 D:\TTTH_KHTN\Python270\Buo107
 - Bai12_Tr28.py
 - Bai13_Tr21.py
 - Bai15_Tr29.py
 - Bai23_Tr30.py
 - temp.py
- External Libraries
- Scratches and Consoles

temp.py Bai13_Tr21.py

```
1 n=int(input("Nhập số"))
2 CauA = lambda n: abs(n)
3 print("Câu A: Trị tuyệt đối của %d là %d" %(n,CauA(n)))
4 -----
5 CauB = lambda n: n+15
6 print("Câu B: %d +15 = %d" %(n,CauB(n)))
7
8
9
10
```

Bai15_Tr29 × Bai23_Tr30 × Bai13_Tr21 ×

2: Favorites

Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]

Nhập số>? -7

Câu A: Trị tuyệt đối của -7 là 7

BuoI07 > Bai13_Tr21.py

1: Project

Project ▾ Buoi07 D:\TTTH_KHTN\Python:
 Bai12_Tr28.py
 Bai13_Tr21.py
 Bai15_Tr29.py
 Bai23_Tr30.py
 temp.py
External Libraries
Scratches and Consoles

temp.py × Bai13_Tr21.py ×

```
1 n=int(input("Nhập số"))
2 CauA = lambda n: abs(n)
3 print("Câu A: Trị tuyệt đối của %d là %d" %(n,CauA(n)))
4 #-----
5 CauB = lambda n: n+15
6 print("Câu B: %d +15 = %d" %(n,CauB(n)))
7 #-----
8 CauC = lambda x, y : x * y
9 x,y = map(int,input("Nhập x, y: ").split(' '))
10 print("CauC: %d * %d = %d"%(x,y,CauC(x,y)))
11
12
13
```

2: Favorites Structure

Bai15_Tr29 × Bai23_Tr30 × Bai13_Tr21 ×

Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]>

Nhập số? -7

Câu A: Trị tuyệt đối của -7 là 7

In[3]:

1: Project

temp.py X Bai13_Tr21.py X

```
7     print("Câu B: %d +15 = %d" %(n,CauB(n)))
8     -----
9     CauC = Lambda x, y : x * y
10    x,y = map(int,input("Nhập x, y: ").split(' '))
11    print("CâuC: %d * %d = %d"%(x,y,CauC(x,y)))'''
12    -----
13    CauD = lambda n: print('%d là bội số của 13 hoặc 19'%(n)) if n%13==0 or n%19==0 else print('không phải bội số')
14    n = int(input("Moi nhập n:"))
15    CauD(n)
```

lambda (n)

Bai15_Tr29 X Bai23_Tr30 X Bai13_Tr21 X Bai13_Tr21 (1) X

Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]

moi nhap n:>? 57

57 là bội số của 13 hoặc 19

In[3]:

n = {int} 57

Special Variables

2: Favorites

Structure

3: Help

PyCharm 2021.3.2 available

temp.py X Bai13_Tr21.py X

```
7     print("Câu B: %d +15 = %d" %(n,CauB(n)))  
8     #-----  
9     CauC = Lambda x, y : x * y  
10    x,y = map(int,input("Nhập x, y: ").split(' '))  
11    print("CauC: %d * %d = %d"%(x,y,CauC(x,y)))'''  
12    #-----  
13    CauD = lambda n: str(n) +' là bội số của 13 hoặc 19' if n%13==0 or n%19==0 else str(n) +'không phải bội số'  
14    n = int(input("mời nhập n: "))  
15    print(CauD(n))
```

có thể viết lại như sau sử
dụng str(n) +

I

File Edit View Navigate Code Refactor Run Tools VCS Window Help Buoi07 [D:\TTTH_KHTN\Python270\Buoi07] - ...\\Bai13_Tr21.py - PyCharm (Adm)

Buoi07 > Bai13_Tr21.py

temp.py Bai13_Tr21.py

1: Project

```
12 #-----
13 CauD = Lambda n: str(n) + ' La boi so cua 13 hoac 19' if n%13==0 or n%19==0 else str(n)
14 n = int(input("moi nhap n:"))
15 print(CauD(n))
16 import math
17 r = float(input('Nhập bán kính hình tròn: '))
18 CauE = lambda r: (math.pi*r*r, 2*math.pi*r)      để sử dụng pi thì phải import math
19 print('Câu E: Diện tích là: %.2f và Chu vi là: %.2f' %(CauE(r)))
```

Buoi07 > Bai13_Tr21.py

te

1: Project

temp.py < Bai13_Tr21.py <

```
15 CauD = Lambda n: str(n) + ' la boi so cua 13 hoac 19' if n%13==0 or n%19==0 else str(n) + 'khong phai boi so'
16 n = int(input("moi nhap n:"))
17 print(CauD(n))'''
```

import math

#CauE = Lambda lu: (math.pi*lu*lu, 2*math.pi*lu)

ki = float(input('Nhập bán kính hình tròn: ')) hoặc có thể đặt tên đối số khác

#print('Câu E: Diện tích là: %.2f và Chu vi là: %.2f' %(CauE(ki)))

CauE = lambda r: print('Câu E: Diện tích là: %.2f và Chu vi là: %.2f' %(math.pi*r*r, 2*math.pi*r))

CauE(ki)

Bai15_Tr29 < Bai23_Tr30 < Bai13_Tr21 < Bai13_Tr21 (1) < temp <

```
import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['D:\\TTTH_KHTN\\Python270\\Buoi07', 'D:/TTTH_KHTN/Python270/Buoi07'])
```

01 ki = {float} 4.0
Special Variables

Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]

(1) Nhập bán kính hình tròn: >? 4

Câu E: Diện tích là: 50.27 và Chu vi là: 25.13

In[3]:

I

```
25     d, r = map(float, input('Nhập chiều dài và chiều rộng hcn cách nhau bởi dấu cách: ')).split()  
26     CauF = lambda d, r: (2*(d+r), d*r)  
27     print('Câu F: chu vi là: %.2f và Diện tích là: %.2f' %(CauF(d, r)))
```

29

```
import math
```

30

```
CauG = lambda n: '%d la so chinh phuong' % (n) if int(math.sqrt(n))**2==n else 'khong phai so chinh phuong'
```

31

```
n = int(input('Nhập n: '))
```

32

```
print(CauG(n))
```

```
34 CauH = lambda a,b,c: '3 tam giac hop le' if (a+b>c or a+c>b or b+c>a) else '3 canh tam giac ko hop le'
35 a,b,c = map(int,input("Nhap 3 canh tam giac: ").split(' '))
36 print(CauH(a,b,c))
```

16/. Hàm nhận 3 tham số là số nguyên (a, b, c). Cho biết a, b, c có là 3 cạnh hợp lệ của 1 tam giác hay không? Nếu là 3 cạnh hợp lệ của tam giác, cho biết đó tam giác gì? (thường, cân, đều, vuông, ...).

temp.py

Bai13_Tr21.py

Bai16_Tr22.py

```
1 # Cau16trg22
2 def Ktra3Canh(a,b,c):
3     if (a + b > c or a + c > b or b + c > a):
4         return True
5     else:
6         return False
7
8 def KtraTamGiac(a,b,c):
9     if (Ktra3Canh(a,b,c)):
10        if(a==b==c):
11            print("Tam giac deu")
12        elif (a==b or a==c or b==c):
13            print("tam giac can")
```

temp.py

Bai13_Tr21.py

Bai16_Tr22.py

```

7
8     def KtraTamGiac(a,b,c):
9         if (Ktra3Canh(a,b,c)):
10            Can= a==b or a==c or b==c    # kết quả là True hoặc False
11            Vuong=(a**2==b**2+c**2 or b**2==a**2+c**2 or c**2==b**2+a**2)
12            if(a==b==c):
13                print("Tam giac deu")
14            elif Can and Vuong:
15                print("tam giac vuong can")
16            elif Can:
17                print("tam giac can")
18            elif Vuong:
19                print("tam giac vuong")
20            else:
21                print("tam giac thuong")

22
23 Cau16 = lambda a,b,c: KtraTamGiac(a,b,c)
24 a,b,c = map(int,input("Nhập 3 cạnh tam giác: ").split(' '))
25 Cau16(a,b,c)

```

17. Viết chương trình cho nhập 1 chuỗi S, lần lượt sử dụng lambda để xét các trường hợp sau:

a. Chuỗi S có bắt đầu bằng ký tự 'P' (in hoa) hay không?

b. Chuỗi S có bắt đầu bằng ký tự 'P' (in hoa) hoặc 'p' (in thường) hay không?

 Gợi ý: thực hiện câu a và b bằng cả 2 cách sau:

- Cách 1 - chuỗi `sFind` chỉ có duy nhất 1 ký tự: `strObject[0] == strFind`.
- Cách 2 - chuỗi `sFind` có thể là 1 hoặc nhiều ký tự: sử dụng phương thức `strObject.startswith(strFind)`. Phương thức này sẽ trả về kết quả là `True` hoặc `False`.

c. Chuỗi S có bắt đầu bằng 1 nguyên âm hay không?

d. Chuỗi S có phải là chuỗi *palindrome* hay không? Một chuỗi được gọi là *palindrome* khi các ký tự trong chuỗi đối xứng nhau.

Ví dụ: chuỗi sau là chuỗi *palindrome*: ABLE WAS I ERE I SAW ELBA

Buoi07 > Bai17_Tr22.py

Bai17_Tr22

temp.py × Bai13_Tr21.py × Bai16_Tr22.py × Bai17_Tr22.py ×

1: Project

```
1
2     #-----
3 S1="Python"
4 S2="python"
5 S3="Hello"
6 print("Câu A:")
7 cau_a11 = lambda S: S+' bắt đầu bằng ký tự P' if s[0].isupper()=='P' else S+' KHÔNG bắt đầu bằng ký tự P'
8 print('Câu A: - cách 1: ', cau_a11(S1))
9
```

1: Project

temp.py × Bai13_Tr21.py × Bai16_Tr22.py × Bai17_Tr22.py ×

```

1
2  -----
3  S1="Python"
4  S2="python"
5  S3="Hello"
6  print("Câu A:")
7  cau_a11 = lambda S: S+ ' bắt đầu bằng ký tự P' if S[0].isupper()=='P' else S+ ' KHÔNG bắt đầu bằng ký tự P'
8  print('Câu A: - cách 1: ', cau_a11(S1))
9
10

```

đoán: typo

lambda (S)

Bai15_Tr29 × Bai23_Tr30 × Bai13_Tr21 × Bai13_Tr21 (1) × temp × temp (1) × Bai17_Tr22 ×

2: Favorites

↻ ⌛ Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]

▶ Câu A:

▶ Câu A: - cách 1: Python KHÔNG bắt đầu bằng ký tự P

In[3]:

01 S1 = {str} 'Python'
01 S2 = {str} 'python'
01 S3 = {str} 'Hello'
> Special Variables

Buoi07 > Bai17_Tr22.py

temp.py × Bai13_Tr21.py × Bai16_Tr22.py × Bai17_Tr22.py ×

```
1
2
3 S1="Python"
4 S2="python"
5 S3="Hello"
6 print("Câu A:")
7 cau_a11 = lambda S: S+' bắt đầu bằng ký tự P' if S[0].upper()=='P' else S+' KHÔNG bắt đầu bằng ký tự P'
8 print('Câu A: - cách 1: ', S1, cau_a11(S1))
9 print('Câu A: - cách 1: ', S3, cau_a11(S1))
10
```

Bai15_Tr29 × Bai23_Tr30 × Bai13_Tr21 × Bai13_Tr21 (1) × temp × temp (1) × Bai17_Tr22 ×

sys.path.extend(['D:\\TTTH_KHTN\\Python270\\Buoi07', 'D:/TTTH_KHTN/Python270/Buoi07'])

Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]

Câu A:

Câu A: - cách 1: Python Python bắt đầu bằng ký tự P

Câu A: - cách 1: Hello Python bắt đầu bằng ký tự P

❶ S1 = {str} 'Python'

❶ S2 = {str} 'python'

❶ S3 = {str} 'Hello'

Special Variables

In[3]:

Buoi07 > Bai17_Tr22.py

Bai17_

1: Project

-

temp.py × Bai13_Tr21.py × Bai16_Tr22.py × Bai17_Tr22.py ×

```
5 S3="Hello"
6 print("Câu A:")
7 cau_a11 = lambda S: S+' bắt đầu bằng ký tự P' if S[0].upper()=='P' else S+' KHÔNG bắt đầu bằng ký tự P'
8 print('Câu A: - cách 1: ', cau_a11(S1))
9 print('Câu A: - cách 1: ', cau_a11(S3))
10 cau_a12 = lambda S: S+' bắt đầu bằng ký tự P' if S.find('P')==0 else S+' KHÔNG bắt đầu bằng ký tự P'
11 print('Câu A: - cách 2: ', cau_a12(S1))
12 print('Câu A: - cách 2: ', cau_a12(S3))
```

13

14

lambda (S)

Bai15_Tr29 × Bai23_Tr30 × Bai13_Tr21 × Bai13_Tr21 (1) × temp × temp (1) × Bai17_Tr22 ×

Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]

Câu A:

Câu A: - cách 1: Python bắt đầu bằng ký tự P

Câu A: - cách 1: Hello KHÔNG bắt đầu bằng ký tự P

Câu A: - cách 2: Python bắt đầu bằng ký tự P

Câu A: - cách 2: Hello KHÔNG bắt đầu bằng ký tự P

① S1 = {str} 'Python'

① S2 = {str} 'python'

① S3 = {str} 'Hello'

> Special Variables

2: Favorites

In[3]:

Structure

Buo107 > Bai17_Tr22.py

Bai1

1: Project

```
temp.py × Bai13_Tr21.py × Bai16_Tr22.py × Bai17_Tr22.py ×  
10  
11     cau_a12 = lambda S: S+' bắt đầu bằng ký tự P' if S.find('P')==0 else S+' KHÔNG bắt đầu bằng ký tự P'  
12     print('Câu A: - cách 2: ', cau_a12(S1))  
13     print('Câu A: - cách 2: ', cau_a12(S3))  
14  
15     cau_a13 = lambda S: S+' bắt đầu bằng ký tự P' if S.startswith('P') else S+' KHÔNG bắt đầu bằng ký tự P'  
16     print('Câu A: - cách 3: ', cau_a13(S1))  
17     print('Câu A: - cách 3: ', cau_a13(S3))  
18
```

lambda (S)

Bai15_Tr29 × Bai23_Tr30 × Bai13_Tr21 × Bai13_Tr21 (1) × temp × temp (1) × Bai17_Tr22 ×

2: Favorites
lecture

- Câu A:
- Câu A: - cách 1: Python bắt đầu bằng ký tự P
- Câu A: - cách 1: Hello KHÔNG bắt đầu bằng ký tự P
- Câu A: - cách 2: Python bắt đầu bằng ký tự P
- Câu A: - cách 2: Hello KHÔNG bắt đầu bằng ký tự P
- Câu A: - cách 3: Python bắt đầu bằng ký tự P
- Câu A: - cách 3: Hello KHÔNG bắt đầu bằng ký tự P

In[3]:

- 01 S1 = {str} 'Python'
- 01 S2 = {str} 'python'
- 01 S3 = {str} 'Hello'

> Special Variables

temp.py × Bai13_Tr21.py × Bai16_Tr22.py × Bai17_Tr22.py ×

```
17 print('Câu A: - cách 3: ', cau_a13(S3))
18 #-----
19 print("Câu B:")
20 cau_b11 = lambda S: S+' bắt đầu bằng ký tự P' if S[0] in 'Pp' else S+' KHÔNG bắt đầu bằng ký tự P'
21 print('Câu B: - cách 1: ', cau_b11(S2))
22 print('Câu B: - cách 1: ', cau_b11(S3))
23
24 cau_b12 = lambda S: S+' bắt đầu bằng ký tự P' if S.find('P')==0 or S.find('p')==0 else S+' KHÔNG bắt đầu bằng ký tự P'
25 print('Câu B: - cách 2: ', cau_b12(S2))
26 print('Câu B: - cách 2: ', cau_b12(S3))
27
28 cau_b13 = lambda S: S+' bắt đầu bằng ký tự P' if S.startswith('P') or S.startswith('p') else S+' KHÔNG bắt đầu bằng ký tự P'
29 print('Câu B: - cách 2: ', cau_b13(S2))
30 print('Câu B: - cách 2: ', cau_b13(S3))
31
```



temp.py × Bai13_Tr21.py × Bai16_Tr22.py × Bai17_Tr22.py ×

```
17 print('Câu A: - cách 3: ', cau_a13(S3))
18 #-----
19 print("Câu B:")
20 cau_b11 = lambda S: S+' bắt đầu bằng ký tự P' if S[0] in 'Pp' else S+' KHÔNG bắt đầu bằng ký tự P'
21 print('Câu B: - cách 1: ', cau_b11(S2))
22 print('Câu B: - cách 1: ', cau_b11(S3))
23
24 cau_b12 = lambda S: S+' bắt đầu bằng ký tự P' if S.find('P') == 0 or S.find('p') == 0 else S+' KHÔNG bắt đầu bằng ký tự P'
25 print('Câu B: - cách 2: ', cau_b12(S2))
26 print('Câu B: - cách 2: ', cau_b12(S3))
27
28 cau_b13 = lambda S: S+' bắt đầu bằng ký tự P' if S.startswith('P') or S.startswith('p') else S+' KHÔNG bắt đầu bằng ký tự P'
29 print('Câu B: - cách 2: ', cau_b13(S2))
30 print('Câu B: - cách 2: ', cau_b13(S3))
```

Bai15_Tr29 × Bai23_Tr30 × Bai13_Tr21 × Bai13_Tr21 (1) × temp × temp (1) × Bai17_Tr22 ×

Câu A: - cách 3: Hello KHÔNG bắt đầu bằng ký tự P

01 S1 = {st
01 S2 = {st

i PyCharm 2021.2.3 available
Update...

```
33 #-----  
34 print("câu c:")  
35 nguyenam='aeiou'  
36 cau_C = lambda s: s+' bắt đầu bằng nguyên âm' if s[0].lower() in nguyenam else s+' KHÔNG bắt đầu bằng nguyên âm'  
37 print(cau_C(S3))  
38 print(cau_C(S4))
```

temp.py × Bai13_Tr21.py × Bai16_Tr22.py × Bai17_Tr22.py ×

```
1
2
3 S1="Python"
4 S2="python"
5 S3="Hello"
6 S4="Apple"
7 S5="madam"← thêm biến này để test thử câu cuối
8 print("Câu A:")
9 cau_a11 = lambda S: S+' bắt đầu bằng ký tự P' if S[0].upper()=='P' else S+' KHÔNG bắt đầu bằng ký tự P'
10 print('Câu A: - cách 1: ', cau_a11(S1))
11 print('Câu A: - cách 1: ', cau_a11(S3))

13 cau_a12 = lambda S: S+' bắt đầu bằng ký tự P' if S.find('P')==0 else S+' KHÔNG bắt đầu bằng ký tự P'
14 print('Câu A: - cách 2: ', cau_a12(S1))
15 print('Câu A: - cách 2: ', cau_a12(S3))

17 cau_a13 = lambda S: S+' bắt đầu bằng ký tự P' if S.startswith('P') else S+' KHÔNG bắt đầu bằng ký tự P'
18 print('Câu A: - cách 3: ', cau_a13(S1))
19 print('Câu A: - cách 3: ', cau_a13(S3))
```

```
41  
42     print("câu D:")  
43     cau_C = lambda S: S+' đối xứng' if S==S[::-1] else S+' KHÔNG đối xứng'  
44     print(cau_C(S3))  
45     print(cau_C(S5))  
46
```

↶ ↷ Câu B: - cách 2: python bắt đầu bằng ký tự P

⤵ ⤶ Câu B: - cách 2: Hello KHÔNG bắt đầu bằng ký tự P

▶ ⏪ Câu C:

⌚ ⌈ Hello KHÔNG bắt đầu bằng nguyên âm

⌚ ⌈ Apple bắt đầu bằng nguyên âm

+ Câu D:

Hello KHÔNG đối xứng

madam đối xứng

In[3]: |

3.7. Xây dựng hàm ẩn danh (Anonymous Function) cho Iterator object

3.7.1. Xây dựng hàm ẩn danh trên number list

86/. Cho 2 list chứa các số nguyên và 2 list này có cùng số lượng phần tử. Sử dụng lambda để tạo ra 1 list mới bằng cách cộng đôi một các số có trong 2 list.

☞ Gói ý: sử dụng hàm map

Ví dụ: `lst1 = [1, 2, 3], lst2 = [4, 5, 6]` => in ra màn hình `[5, 7, 9]`

<code>lst1 = [1, 2, 3]</code>
<code>lst2 = [4, 5, 6]</code>
<code>resultList = [5, 7, 9]</code>

4.7.2. Một số ví dụ về dùng 2 hàm map và filter trên iterator object

4.7.2.1. Ví dụ so sánh cách dùng 2 hàm map và filter

- **Ví dụ 4.102** minh họa số lượng phần tử của 2 tập kết quả bằng nhau khi sử dụng hàm *map* và *filter*: Cho list $L=[1, 2, 3, 4]$, sử dụng hàm square để tạo ra list mới (*lst*) chứa bình phương các phần tử có trong L .



Sử dụng hàm thông thường	Sử dụng Anonymous function	Kết quả
<pre>def square(x): return x**2</pre>		
<pre>L=[1,2,3,4] lst=list(map(square, L)) print('map:',lst)</pre>	<pre>L=[1,2,3,4] lst=list(map(lambda x: x**2, L)) print('map:',lst)</pre>	map: [1, 4, 9, 16]
<pre>lst=list(filter(square, L)) print('filter:',lst)</pre>	<pre>lst=list(filter(lambda x: x**2, L)) print('filter:',lst)</pre>	filter: [1, 2, 3, 4]

- **Ví dụ 4.104** minh họa việc dựa vào mục đích mà người lập trình mong muốn về tập kết quả để chọn sử dụng hàm map hay filter: Cho list $L = [1, 3, 2, 5, 20, 21]$, tạo ra list mới (lst) với biểu thức trong lambda là $x \% 2 == 0$ (phần tử trong L là số chẵn).

Sử dụng Anonymous function

```
L=[1, 3, 2, 5, 20, 21]
lst=list(map(lambda x: x % 2 == 0, L))
print('map:', lst)
```

Kết quả

```
map: [False, False, True, False,
      True, False]
```

```
L=[1, 3, 2, 5, 20, 21]
lst=list(filter(lambda x: x % 2 == 0, L))
print('filter:', lst)
```

```
filter: [2, 20]
```

- **Ví dụ 4.108:** cho 2 list lst1 và lst2 chứa các số nguyên. Cho biết các số có trong cả 2 list.

Sử dụng hàm thông thường

```
def intersect(L1,L2):  
    L3=[ ]  
    for i in L1:  
        if i in L2:  
            L3.append(i)  
    return L3
```

Sử dụng Anonymous function

```
def intersect(L1,L2):  
    return list(filter(lambda x: num in  
                      x in lst2 , lst1))
```

Kết quả

[2, 3, 5]

4.7.2.4. Có thể sử dụng 1 trong 2 hàm map và filter để giải quyết cùng 1 vấn đề

- **Ví dụ 4.111:** cho 1 list chứa số nguyên. Yêu cầu đếm số lượng số chẵn, số lẻ trong list

Mã lệnh

Kết quả

<pre>lst=[4, 9, 7, 5, 6] print('Dùng hàm map:') slChan=list(map(lambda x: x%2==0, lst)).count(True) slLe=list(map(lambda x: x%2==0, lst)).count(False) print('Trong list có %d số chẵn và %d số lẻ'%(slChan, slLe)) print('Dùng hàm filter:') slChan=len(list(filter(lambda x: x%2 == 0, lst))) slLe=len(list(filter(lambda x: x%2 != 0, lst))) print('Trong list có %d số chẵn và %d số lẻ'%(slChan, slLe))</pre>	<p>Dùng hàm map: Trong list có 2 số chẵn và 3 số lẻ</p> <p>Dùng hàm filter: Trong list có 2 số chẵn và 3 số lẻ</p>
---	--

3.1.5. Sử dụng hàm map hoặc filter trên list

34. Cho list lst chứa số nguyên. Sử dụng hàm map để các số nguyên trong lst thành dạng chuỗi

Ví dụ: Original list: [1, 2, 3, 4]

List of strings: ['1', '2', '3', '4']

Project ▾ + - temp.py × Bai13_Tr21.py × Bai16_Tr22.py × Bai17_Tr22.py × Bai34_Tr33.py ×

Buoi07 D:\TTTH_KHTN\Python\Buoi07

- Bai12_Tr28.py
- Bai13_Tr21.py
- Bai15_Tr29.py
- Bai16_Tr22.py
- Bai17_Tr22.py
- Bai23_Tr30.py
- Bai34_Tr33.py**
- temp.py

```

1 lst1=[1,2,3]
2 print("List 1=",lst1)
3 resultList= list(map(str,lst1))
4 print("ResultList=", resultList)

```

External Libraries

Scratches and Consoles

Bai15_Tr29 × Bai23_Tr30 × Bai13_Tr21 × Bai13_Tr21 (1) × temp × temp (1) × Bai17_Tr22 × Bai34_Tr33 ×

sys.path.extend(['D:\\TTTH_KHTN\\Python270\\Buoi07', 'D:/TTTH_KHTN/Python270/Buoi07'])



Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AM)]

List 1= [1, 2, 3]

ResultList= ['1', '2', '3']

In[3]:

35/. Cho 2 list B và E có số lượng phần tử bằng nhau. Sử dụng hàm map để tạo ra 1 list mới với kết quả là B^E của từng cặp phần tử

Ví dụ: bases_num = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

exponential = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

result= [10, 400, 27000, 2560000, 312500000, 46656000000, 8235430000000,
1677721600000000, 38742048900000000, 10000000000000000]

BuoI07 > Bai35_Tr33.py

Bai35_Tr33

Project

1: Project

BuoI07 D:\TTTH_KHTN\Python

- temp.py
- Bai13_Tr21.py
- Bai16_Tr22.py
- Bai17_Tr22.py
- Bai34_Tr33.py
- Bai35_Tr33.py

1 import math
2 lstB=list(range(10, 101,10))
3 lstE=list(range(1, 11))
4 print(lstB)
5 print(lstE)
6 resultList=list(map(pow,lstB,lstE))
7 print(resultList)

External Libraries

Scratches and Consoles

Bai15_Tr29 × Bai23_Tr30 × Bai13_Tr21 × Bai13_Tr21 (1) × temp × temp (1) × Bai17_Tr22 × Bai34_Tr33 × Bai35_Tr33 ×

2 Favorites

Structure

In[2]: runfile('D:/TTTH_KHTN/Python270/Bu...')
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
In[3]: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
In[4]: [10, 400, 27000, 2560000, 31250000, 46656000000, 8235430000000, 16777216000000000, 3874204890000000000000000]

36/. Cho 2 list $lstX$ và $lstY$ chứa các số nguyên và có cùng số lượng phần tử.

- Viết hàm *Addition_Subtraction* với tham số truyền cho hàm là 2 số nguyên x và y . Hàm trả về 2 giá trị nguyên là $x-y$ và $x+y$.
- Trong chương trình chính, sử dụng hàm map để tạo ra 1 list mới chứa các bộ 2 số do hàm *Addition_Subtraction* trả về khi nhận các giá trị của 2 list $lstX$ và $lstY$.

Ví dụ: với 2 list: $listX = [6, 5, 3, 9]$ và $listY = [0, 1, 7, 7]$

Sẽ xuất ra: Kết quả thực hiện: $[(6, 6), (6, 4), (10, -4), (16, 2)]$

Buoi07 > Bai36_Tr34.py

Project - temp.py × Bai13_Tr21.py × Bai16_Tr22.py × Bai17_Tr22.py × Bai34_Tr33.py × Bai35_Tr33.py × Bai36_Tr34.py

1: Project Buoi07 D:\TTTH_KHTN\Python\ 1 def Cong_Tru(a,b):
2 return a+b, a-b
3 lstX=[6,5,3,9]
4 lstY=[0,1,7,7]
5 print(lstX)
6 print(lstY)
7 resultList=list(map(Cong_Tru, lstX, lstY))
8 print(resultList)

> External Libraries
> Scratches and Consoles

Cong_Tru()

Bai15_Tr29 × Bai23_Tr30 × Bai13_Tr21 × Bai13_Tr21 (1) × temp × temp (1) × Bai17_Tr22 × Bai34_Tr33 × Bai35_Tr33 × Bai36_Tr34 ×

2: Favorites

2: Structure

In[3]:

Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]

[6, 5, 3, 9]

[0, 1, 7, 7]

[(6, 6), (6, 4), (10, -4), (16, 2)]

IstX = {list: 4} [6, 5, 3, 9]
IstY = {list: 4} [0, 1, 7, 7]
resultList = {list: 4} [(6, 6), (6, 4), (10, -4), (16, 2)]
Special Variable